

Package ‘bioCancer’

October 14, 2019

Title Interactive Multi-Omics Cancers Data Visualization and Analysis

Version 1.12.0

Date 2018-09-04

Description bioCancer is a Shiny App to visualize and analyse interactively Multi-Assays of Cancer Genomic Data.

Depends R (>= 3.4.0), radiant.data (>= 0.9.1), cgdsr(>= 1.2.6), XML(>= 3.98)

Imports DT (>= 0.3), dplyr (>= 0.7.2), shiny (>= 1.0.5), AlgDesign (>= 1.1.7.3), import (>= 1.1.0), methods, shinythemes, Biobase, geNetClassifier, AnnotationFuncs, org.Hs.eg.db, DOSE, clusterProfiler, reactome.db, ReactomePA, DiagrammeR(>= 0.7), visNetwork, htmlwidgets, plyr, tibble

Suggests BiocStyle, rmarkdown, knitr, testthat (>= 0.10.0)

VignetteBuilder knitr

URL <http://kmezhoud.github.io/bioCancer>

BugReports <https://github.com/kmezhoud/bioCancer/issues>

License AGPL-3 | file LICENSE

LazyData true

biocViews GUI, DataRepresentation, Network, MultipleComparison, Pathways, Reactome, Visualization, GeneExpression, GeneTarget

RoxygenNote 6.1.1

Encoding UTF-8

git_url <https://git.bioconductor.org/packages/bioCancer>

git_branch RELEASE_3_9

git_last_commit 252f51e

git_last_commit_date 2019-05-02

Date/Publication 2019-10-13

Author Karim Mezhoud [aut, cre]

Maintainer Karim Mezhoud <kmezhoud@gmail.com>

R topics documented:

attriColorGene	3
attriColorValue	3
attriColorVector	4
attriShape2Gene	5
attriShape2Node	5
bioCancer	6
checkDimensions	6
coffeewheel	7
coffeewheelOutput	8
displayTable	8
Edges_Diseases_obj	9
epiGenomics	10
findPhantom	10
getFreqMutData	11
getGenesClassification	11
getListProfData	12
getList_Cases	13
getList_GenProfs	14
getMegaProfData	14
getSequenced_SampleSize	15
grepRef	16
metabologram	17
metabologramOutput	18
Mutation_obj	18
Node_df_FreqIn	19
Node_Diseases_obj	20
Node_obj_CNA_ProfData	20
Node_obj_FreqIn	21
Node_obj_Met_ProfData	22
Node_obj_mRNA_Classifier	22
renderCoffeewheel	23
renderMetabologram	24
reStrColorGene	24
reStrDimension	25
reStrDisease	26
returnTextAreaInput	26
Studies_obj	27
switchButton	28
UnifyRowNames	28
user_CNA	29
user_MetHM27	29
user_MetHM450	30
user_mRNA	30
user_Mut	31
whichGeneList	31
widgetThumbnail	32
Index	33

attriColorGene *Attribute Color to Gene*

Description

Attribute Color to Gene

Usage

```
attriColorGene(df)
```

Arguments

df data frame with mRNA or CNA or mutation frequency or methylation (numeric).

Value

A list colors for every gene

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
ProfData <- getProfileData(cgds,
  geneList, "gbm_tcga_pub_mrna", "gbm_tcga_pub_all")
rownames(ProfData) <- NULL
clr <- attriColorGene(ProfData)

## End(Not run)
```

attriColorValue *Attribute Color to Value*

Description

Attribute Color to Value

Usage

```
attriColorValue(Value, df, colors=c(a,b,c),feet)
```

Arguments

Value integer
df data frame with numeric values
colors a vector of 5 colors
feet the interval between two successive colors in the palette (0.1)

Value

Hex Color Code

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
ProfData <- getProfileData(cgds,
  geneList, "gbm_tcga_pub_mrna", "gbm_tcga_pub_all")
rownames(ProfData) <- NULL
clrRef <- attriColorValue(1.2,
  ProfData,
  colors = c("blue3", "white", "red"),
  feet=10)

## End(Not run)
```

attriColorVector *Attribute color to a vector of numeric values*

Description

Attribute color to a vector of numeric values

Usage

```
attriColorVector(Value, vector, colors=c(a,b,c),feet)
```

Arguments

Value	numeric
vector	A vector of numeric data
colors	3 colors
feet	An interval between two numeric value needed to change the color

Value

A vetor of colors

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
ProfData <- getProfileData(cgds,
  geneList, "gbm_tcga_pub_mrna", "gbm_tcga_pub_all")
rownames(ProfData) <- NULL
clrVec <- attriColorVector(1.2,
  ProfData[1,],
  colors = c("blue", "white", "red"),
```

```
    feet=1)
## End(Not run)
```

attriShape2Gene *Attribute shape to nodes*

Description

Attribute shape to nodes

Usage

```
attriShape2Gene(gene, genelist)
```

Arguments

gene	Gene symbol
genelist	Gene list

Value

A character "BRCA1[shape = 'circle', "

Examples

```
how <- "runManually"
## Not run:
GeneList <- whichGeneList("73")
attriShape2Gene("P53", GeneList)
attriShape2Gene("GML", GeneList)

## End(Not run)
```

attriShape2Node *Attributes shape to Nodes*

Description

Attributes shape to Nodes

Usage

```
attriShape2Node(gene, genelist)
```

Arguments

gene	symbol "TP53"
genelist	a vector of gene symbol

Value

A data frame with egdes attributes

Examples

```
GeneList <- c("DKK3" , "NBN" , "MY06" , "TP53" , "PML" , "IFI16" ,"BRCA1")
NodeShape <- attriShape2Gene("DKK3", GeneList)
```

bioCancer

Launch bioCancer with default browser

Description

Launch bioCancer with default browser

Usage

```
bioCancer()
```

Value

web page of bioCancer Shiny App

Examples

```
ShinyApp <- 1
## Not run:
bioCancer()

## End(Not run)
```

checkDimensions

Check wich Cases and genetic profiles are available for every seleted study

Description

Check wich Cases and genetic profiles are available for every seleted study

Usage

```
checkDimensions(panel, StudyID)
```

Arguments

panel panel can take to strings 'Circomics' or 'Networking'
 StudyID Study reference using cgdsr index

Value

A data frame with two column (Cases, Genetic profiles). Every row has a dimension (CNA, mRNA...). The data frame is filled with yes/no response.

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
df <- checkDimensions(panel='Networking', StudyID= "gbm_tcga_pub")

## End(Not run)
```

coffeewheel	<i>This is an htmlwidgets-based visualization tool for hierarchical data. It is zoomable, meaning that you can interact with the hierarchy and zoom in/out accordingly.</i>
-------------	---

Description

This is an htmlwidgets-based visualization tool for hierarchical data. It is zoomable, meaning that you can interact with the hierarchy and zoom in/out accordingly.

Usage

```
coffeewheel(treeData, width=600, height=600, main="", partitionAttribute="value")
```

Arguments

treeData	A hierarchical tree data as in example
width	600
height	600
main	Title
partitionAttribute	"value"

Value

A circular layout with genetic profile.

Examples

```
How <- "runManually"
## Not run:
coffeewheel(treeData = sampleWheelData)

## End(Not run)
```

coffeewheelOutput	<i>Widget output function for use in Shiny</i>
-------------------	--

Description

Widget output function for use in Shiny

Usage

```
coffeewheelOutput(outputId, width=700, height=700)
```

Arguments

outputId	id
width	700
height	700

Value

A circular layout with genetic profile in Shiny App.

Examples

```
How <- "runManually"  
## Not run:  
coffeewheel(treeData = sampleWheelData)  
  
## End(Not run)
```

displayTable	<i>Display dataframe in table using DT package</i>
--------------	--

Description

Display dataframe in table using DT package

Usage

```
displayTable(df)
```

Arguments

df	a dataframe
----	-------------

Value

A table

Examples

```

session <- NULL
cgds <- CGDS("http://www.cbioportal.org/")
Studies<- getCancerStudies(cgds)
## Not run:
displayTable(Studies)

## End(Not run)

```

Edges_Diseases_obj *get Edges dataframe for Gene/Disease association from geNetClassifier*

Description

get Edges dataframe for Gene/Disease association from geNetClassifier

Usage

```
Edges_Diseases_obj(genesclassdetails)
```

Arguments

genesclassdetails
a dataframe from geNetClassifier

Value

A data frame with egdes attributes

Examples

```

GenesClassDetails <- structure(list(Genes = c("FANCF", "MLH1", "MSH2", "ATR", "PARP1",
"CHEK2", "RAD51"), ranking = c(1L, 1L, 1L, 2L, 3L, 1L, 2L), class = c("brca_tcga",
"gbm_tcga", "lihc_tcga", "lihc_tcga", "lihc_tcga", "lusc_tcga",
"lusc_tcga"), postProb = c(1, 0.99, 1, 0.99, 0.99, 1,
0.98), exprsMeanDiff = c(180, 256, -373, -268,
-1482, 258, 143), exprsUpDw = c("UP", "UP", "DOWN",
"DOWN", "DOWN", "UP", "UP")), .Names = c("Genes", "ranking",
"class", "postProb", "exprsMeanDiff", "exprsUpDw"),
class = "data.frame", row.names = c(NA,-7L))

```

```
Ed_Diseases_obj <- Edges_Diseases_obj(genesclassdetails=GenesClassDetails)
```

epiGenomics	<i>Default dataset of bioCancer</i>
-------------	-------------------------------------

Description

Default dataset of bioCancer

Usage

epiGenomics

Format

An object of class `data.frame` with 48 rows and 7 columns.

Author(s)

Karim Mezhoud <kmezhoud@gmail.com>

findPhantom	<i>Check if PhantomJS is installed. Similar to webshot</i>
-------------	--

Description

Check if PhantomJS is installed. Similar to webshot

Usage

findPhantom()

Value

Logic object

Examples

```
How <- "runManually"  
## Not run:  
findPhantom()  
  
## End(Not run)
```

getFreqMutData	<i>get mutation frequency</i>
----------------	-------------------------------

Description

get mutation frequency

Usage

```
getFreqMutData(list, geneListLabel)
```

Arguments

`list` a list of data frame with mutation data. Each data frame is for one study
`geneListLabel` file name of geneList examples: "73"

Value

a data frame with mutation frequency. gene is in rows and study is in column

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")  
## Not run:  
geneList <- whichGeneList("73")  
r_data <- new.env()  
MutData <- getMutationData(cgds,"gbm_tcga_pub_all",  
  "gbm_tcga_pub_mutations", geneList )  
FreqMut <- getFreqMutData(list(ls1=MutData, ls2=MutData), "73")  
  
## End(Not run)
```

getGenesClassification	<i>get genes classification</i>
------------------------	---------------------------------

Description

get genes classification

Usage

```
getGenesClassification(checked_Studies, GeneList,  
  samplesize, threshold, listGenProfs, listCases)
```

Arguments

checked_Studies	checked studies
GeneList	gene list
samplesize	sample size
threshold	p-value threshold
listGenProfs	list of genetic profiles
listCases	list of cases

Value

A table with genes classed by study

Examples

```

cgds <- CGDS("http://www.cbioportal.org/")
listStudies <- cgdsr::getCancerStudies(cgds)
## Not run:
checked_Studies <- listStudies[3:5]
listCases <- getList_Cases(listStudies[1:3])
listGenProfs <- getList_GenProfs(listStudies[1:3])
GeneList <- c('P53', 'IFI16', 'BRCA1')
samplesize <- 50
threshold <- 0.95
table <- getGenesClassification(checked_Studies, GeneList,
samplesize ,threshold ,listGenProfs, listCases)

## End(Not run)

```

getListProfData	<i>get list of data frame with profiles data (CNA,mRNA, Methylation, Mutation...)</i>
-----------------	---

Description

get list of data frame with profiles data (CNA,mRNA, Methylation, Mutation...)

Usage

```
getListProfData(panel, geneListLabel)
```

Arguments

panel	Panel name (string) in which Studies are selected. There are two panels ("Circomics" or "Networking")
geneListLabel	The label of GeneList. There are three cases: "Genes" user gene list, "Reactome_GeneList" GeneList plus genes from reactomeFI "file name" from Examples

Value

A LIST of profiles data (CNA, mRNA, Methylation, Mutation, miRNA, RPPA). Each dimension content a list of studies.

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
r_data <- new.env()
MutData <- cgdsr::getMutationData(cgds,"gbm_tcga_pub_all",
  "gbm_tcga_pub_mutations", geneList )
FreqMut <- getFreqMutData(list(ls1=MutData, ls2=MutData), "73")
input <- NULL
input[['StudiesIDCircos']] <- c("luad_tcga_pub","blca_tcga_pub")

ListProfData <- getListProfData(panel= "Circomics","73")

## End(Not run)
```

getList_Cases

get list of cases of each selected study in Classifier panel

Description

get list of cases of each selected study in Classifier panel

Usage

```
getList_Cases(checked_Studies)
```

Arguments

```
checked_Studies
      checked studies
```

Value

listes of cases

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
listStudies <- cgdsr::getCancerStudies(cgds)
## Not run:
listCases <- getList_Cases(listStudies[1:3])

## End(Not run)
```

getList_GenProfs	<i>get list of genetic profiles of each selected study in Classifier panel</i>
------------------	--

Description

get list of genetic profiles of each selected study in Classifier panel

Usage

```
getList_GenProfs(checked_Studies)
```

Arguments

```
checked_Studies
                checked studies
```

Value

listes of genetics profiles

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
listStudies <- cgdsr::getCancerStudies(cgds)
## Not run:
listGenProfs <- getList_GenProfs(listStudies[1:3])

## End(Not run)
```

getMegaProfData	<i>search and get genetic profiles (CNA,mRNA, Methylation, Mutation...) of gene list upper than 500</i>
-----------------	---

Description

search and get genetic profiles (CNA,mRNA, Methylation, Mutation...) of gene list upper than 500

Usage

```
getMegaProfData(MegaGeneList, GenProf, Case, Class)
```

Arguments

```
MegaGeneList  A list of genes upper than 500
GenProf       genetic profile reference
Case         Case reference
Class        indicates the panel ProfData or Mutdata
```

Details

See <https://github.com/kmezhound/bioCancer/wiki>

Value

A data frame with Genetic profile

Examples

```
GeneList <- c("ALK", "JAK3", "SHC3", "TP53", "MYC", "PARP")
## Not run:
cgds <- cgdsr::CGDS("http://www.cbioportal.org/")
listCase_gbm_tcga_pub <- cgdsr::getCaseLists(cgds, "gbm_tcga_pub")[,1]
listGenProf_gbm_tcga_pub <- cgdsr::getGeneticProfiles(cgds, "gbm_tcga_pub")[,1]

ProfData_Mut <- grepRef("gbm_tcga_pub_all", listCase_gbm_tcga_pub,
  "gbm_tcga_pub_mutations", listGenProf_gbm_tcga_pub, GeneList, Mut=1)

## End(Not run)
```

getSequenced_SampleSize

get samples size of sequenced genes

Description

get samples size of sequenced genes

Usage

```
getSequenced_SampleSize(StudyID)
```

Arguments

StudyID Study reference using cgdsr index

Value

dataframe with sample size for each selected study.

Examples

```
## Not run:
sampleSize <- getSequenced_SampleSize(input$StudiesIDCircos)

## End(Not run)
```

grepRef *search and get genetic profiles (CNA,mRNA, Methylation, Mutation...)*

Description

search and get genetic profiles (CNA,mRNA, Methylation, Mutation...)

Usage

```
grepRef(regex1, listRef1, regex2, listRef2, GeneList, Mut)
```

Arguments

regex1	Case id (cancer_study_id_[mutations, cna, methylation, mrna]).
listRef1	A list of cases for one study.
regex2	Genetic Profile id (cancer_study_id_[mutations, cna, methylation, mrna]).
listRef2	A list of Genetic Profiles for one study.
GeneList	A list of genes
Mut	Condition to set if the genetic profile is mutation or not (0,1)

Details

See <https://github.com/kmezhoud/bioCancer/wiki>

Value

A data frame with Genetic profile

Examples

```
GeneList <- c("ALK", "JAK3", "SHC3", "TP53", "MYC", "PARP")
## Not run:
cgds <- cgdsr::CGDS("http://www.cbioportal.org/")
listCase_gbm_tcga_pub <- cgdsr::getCaseLists(cgds, "gbm_tcga_pub")[,1]
listGenProf_gbm_tcga_pub <- cgdsr::getGeneticProfiles(cgds, "gbm_tcga_pub")[,1]

ProfData_Mut <- grepRef("gbm_tcga_pub_all", listCase_gbm_tcga_pub,
  "gbm_tcga_pub_mutations", listGenProf_gbm_tcga_pub, GeneList, Mut=1)

## End(Not run)
```

metabologram	<i>Circular plot of hierarchital data of genetic profile.</i>
--------------	---

Description

Circular plot of hierarchital data of genetic profile.

Usage

```
metabologram(treeData,width=600,height=600,main="",showLegend=FALSE,
              legendBreaks=NULL,
              legendColors=NULL,
              fontSize=12,
              legendText="Legend")
```

Arguments

treeData	A hierarchical tree data as in example
width	600
height	600
main	Title
showLegend	FALSE
legendBreaks	NULL
legendColors	NULL
fontSize	12
legendText	Legend

Value

A circular layout with genetic profile.

See Also

<https://github.com/armish/metabologram>

Examples

```
How <- "runManually"
## Not run:
metabologram(treeData = sampleWheelData, width=600,
              height=600, main="title", showLegend = TRUE, fontSize = 10,
              legendBreaks=c("NA","Min","Negative", "0", "Positive", "Max"),
              legendColors=c("black","blue","cyan","white","yellow","red") ,
              legendText="Legend")

## End(Not run)
```

metabologramOutput *Widget output function for use in Shiny*

Description

Widget output function for use in Shiny

Usage

```
metabologramOutput(outputId, width = 600, height = 500)
```

Arguments

outputId	id
width	600
height	600

Value

A circular plot with genetic profile in Shiny App.

Examples

```
## Not run:
library(bioCancer)
bioCancer::metabologram(treeData = sampleMetabologramData)

## End(Not run)
```

Mutation_obj *Attribute mutation frequency to nodes*

Description

Attribute mutation frequency to nodes

Usage

```
Mutation_obj(list, FreqMutThreshold, geneListLabel)
```

Arguments

list	A list of data frame with mutation data. Each data frame to study
FreqMutThreshold	threshold Rate of cases (patients) having mutation (0-1).
geneListLabel	file name of geneList examples: "73"

Value

A dat frame with mutation frequency. Ech column corresponds to a study.

Examples

```

cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
MutData <- getMutationData(cgds,"gbm_tcga_pub_all",
"gbm_tcga_pub_mutations", geneList )
listMutData <- list(ls1=MutData, ls2=MutData)
FreqMutThreshold <- 10
r_data <- new.env()
MutObj <- Mutation_obj(listMutData, 10, "73")

## End(Not run)

```

Node_df_FreqIn

*Attributes size to Nodes depending on number of interaction***Description**

Attributes size to Nodes depending on number of interaction

Usage

```
Node_df_FreqIn(geneList, freqIn)
```

Arguments

geneList	a vector of genes
freqIn	dataframe with Node interaction frequencies

Value

A data frame with nodes size attributes

Examples

```

Node_df_FreqIn
## Not run:
r_data <- new.env()
r_data[["FreqIn"]] <- structure(list(Genes = c("ATM", "ATR", "BRCA1", "BRCA2", "CHEK1",
"CHEK2", "FANCF", "MDC1", "RAD51"), FreqSum = c(0.04, 0.05, 0.05,
0.03, 0.05, 0.04, 0.03, 0.03, 0.02)), .Names = c("Genes", "FreqSum"),
class = "data.frame", row.names = c(NA, -9L))
GeneList <- whichGeneList("DNA_damage_Response")
node_df <- Node_df_FreqIn(GeneList, r_data$FreqIn)

## End(Not run)

```

Node_Diseases_obj *Attributes color and shape to Nodes of Diseases*

Description

Attributes color and shape to Nodes of Diseases

Usage

```
Node_Diseases_obj(genesclassdetails)
```

Arguments

```
genesclassdetails
    a dataframe from geNetClassifier function
```

Value

A data frame with nodes Shapes and colors

Examples

```
GenesClassDetails <- structure(list(Genes = c("FANCF", "MLH1", "MSH2", "ATR", "PARP1",
"CHEK2", "RAD51"), ranking = c(1L, 1L, 1L, 2L, 3L, 1L, 2L), class = c("brca_tcga",
"gbm_tcga", "lihc_tcga", "lihc_tcga", "lihc_tcga", "lusc_tcga",
"lusc_tcga"), postProb = c(1, 0.99, 1, 0.99, 0.99, 1,
0.98), exprsMeanDiff = c(180, 256, -373, -268,
-1482, 258, 143), exprsUpDw = c("UP", "UP", "DOWN",
"DOWN", "DOWN", "UP", "UP")), .Names = c("Genes", "ranking",
"class", "postProb", "exprsMeanDiff", "exprsUpDw"),
class = "data.frame", row.names = c(NA,-7L))
Node_Diseases_df <- Node_Diseases_obj(genesclassdetails= GenesClassDetails)
```

Node_obj_CNA_ProfData *Attribute CNA data to node border*

Description

Attribute CNA data to node border

Usage

```
Node_obj_CNA_ProfData(list)
```

Arguments

```
list                      A list of data frame with CNA data. Each data frame corresponds to a study.
```

Value

A data frame with node border attributes

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
GeneList <- whichGeneList("DNA_damage_Response")
ProfDataCNA <- cgdsr::getProfileData(cgds, GeneList, "brca_tcga_pub_gistic", "brca_tcga_pub_all")
ListProfDataCNA <- list(ls1=ProfDataCNA, ls2=ProfDataCNA)
nodeObj <- Node_obj_CNA_ProfData(ListProfDataCNA)

## End(Not run)
```

Node_obj_FreqIn	<i>Attribute interaction frequency to node size</i>
-----------------	---

Description

Attribute interaction frequency to node size

Usage

```
Node_obj_FreqIn(geneList)
```

Arguments

geneList A list of gene symbol

Value

A data frame with node attributes

Examples

```
r_data <- new.env()
r_data[["FreqIn"]] <- structure(list(Genes = c("ATM", "ATR", "BRCA1", "BRCA2", "CHEK1",
"CHEK2", "FANCF", "MDC1", "RAD51"), FreqSum = c(0.04, 0.05, 0.05,
0.03, 0.05, 0.04, 0.03, 0.03, 0.02)), .Names = c("Genes", "FreqSum"),
class = "data.frame", row.names = c(NA, -9L))
## Not run:
GeneList <- whichGeneList("DNA_damage_Response")
nodeObj <- Node_obj_FreqIn(GeneList)

## End(Not run)
```

Node_obj_Met_ProfData *Attribute gene Methylation to Nodes*

Description

Attribute gene Methylation to Nodes

Usage

```
Node_obj_Met_ProfData(list, type, threshold)
```

Arguments

list	a list of data frame with methylation data
type	HM450 or HM27
threshold	the Rate cases (patients) that have a silencing genes by methylation

Value

a data frame with node shape attributes

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
GeneList <- whichGeneList("DNA_damage_Response")
ProfDataMET <- cgdsr::getProfileData(cgds, GeneList, "gbm_tcga_pub_methylation", "gbm_tcga_pub_all")
ListProfDataMET <- list(ls1=ProfDataMET, ls2=ProfDataMET)
nodeObj <- Node_obj_Met_ProfData(ListProfDataMET, "HM450", 0.1)

## End(Not run)
```

Node_obj_mRNA_Classifier

Attribute genes expression to color nodes

Description

Attribute genes expression to color nodes

Usage

```
Node_obj_mRNA_Classifier(geneList, genesclassdetails)
```

Arguments

geneList	A gene list.
genesclassdetails	A dataframe with genes classes and genes expression.

Value

A data frame with node color attributes

Examples

```
r_data <- new.env()
input <- NULL

r_data[["FreqIn"]] <- structure(list(Genes = c("ATM", "ATR", "BRCA1", "BRCA2", "CHEK1",
"CHEK2", "FANCF", "MDC1", "RAD51"), FreqSum = c(0.04, 0.05, 0.05,
0.03, 0.05, 0.04, 0.03, 0.03, 0.02)), .Names = c("Genes", "FreqSum"),
class = "data.frame", row.names = c(NA, -9L))

GenesClassDetails <- structure(list(Genes = c("FANCF", "MLH1", "MSH2", "ATR", "PARP1",
"CHEK2", "RAD51"), ranking = c(1L, 1L, 1L, 2L, 3L, 1L, 2L), class = c("brca_tcga",
"gbm_tcga", "lihc_tcga", "lihc_tcga", "lihc_tcga", "lusc_tcga",
"lusc_tcga"), postProb = c(1, 0.99, 1, 0.99, 0.99, 1,
0.98), exprsMeanDiff = c(180, 256, -373, -268,
-1482, 258, 143), exprsUpDw = c("UP", "UP", "DOWN",
"DOWN", "DOWN", "UP", "UP")), .Names = c("Genes", "ranking",
"class", "postProb", "exprsMeanDiff", "exprsUpDw"),
class = "data.frame", row.names = c(NA,-7L))
## Not run:
GeneList <- whichGeneList("DNA_damage_Response")
nodeObj <- Node_obj_mRNA_Classifier(GeneList, GenesClassDetails)

## End(Not run)
```

renderCoffeewheel

Widget render function for use in Shiny

Description

Widget render function for use in Shiny

Usage

```
renderCoffeewheel(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

expr	id
env	parent.frame
quoted	FALSE

Value

A circular layout with genetic profile in Shiny App.

Examples

```
How <- "runManually"
## Not run:
coffeewheel(treeData = sampleWheelData)

## End(Not run)
```

renderMetabologram	<i>Widget render function for use in Shiny</i>
--------------------	--

Description

Widget render function for use in Shiny

Usage

```
renderMetabologram(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

expr	expression
env	parent.frame()
quoted	FALSE

Value

A circular plot with genetic profile in Shiny App.

Examples

```
## Not run:
library(bioCancer)
bioCancer::metabologram(treeData = sampleMetabologramData)

## End(Not run)
```

reStrColorGene	<i>Restructure the list of color attributed to the genes in every dimenssion for every studies</i>
----------------	--

Description

Restructure the list of color attributed to the genes in every dimenssion for every studies

Usage

```
reStrColorGene(df)
```

Arguments

df	data frame with colors attributed to the genes
----	--

Value

Hierarchical color attribute: gene > color

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
ProfData <- getProfileData(cgds,
  geneList, "gbm_tcga_pub_mrna", "gbm_tcga_pub_all")
rownames(ProfData) <- NULL
ls <- reStrColorGene(ProfData)

## End(Not run)
```

reStrDimension	<i>Restructure the list of color attributed to the genes in every study for every dimensions</i>
----------------	--

Description

Restructure the list of color attributed to the genes in every study for every dimensions

Usage

```
reStrDimension(LIST)
```

Arguments

LIST list of hierarchical dimensions

Value

Hierarchical structure of: Study > dimensions > gene > color

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
ProfData <- getProfileData(cgds,
  geneList, "gbm_tcga_pub_mrna", "gbm_tcga_pub_all")
rownames(ProfData) <- NULL
TREE <- reStrDimension(list(
  list1=list(df1=ProfData,df2=ProfData),
  list2=list(df3=ProfData,df4=ProfData)))

## End(Not run)
```

reStrDisease	<i>Restructure the list of color attributed to the genes in every disease</i>
--------------	---

Description

Restructure the list of color attributed to the genes in every disease

Usage

```
reStrDisease(List)
```

Arguments

List of data frame with color attributes

Value

Hierarchy of dimensions in the same study: dimensions > gene > color

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
ProfData <- getProfileData(cgds,
  geneList, "gbm_tcga_pub_mrna", "gbm_tcga_pub_all")
rownames(ProfData) <- NULL
tree <- reStrDisease(list(df1=ProfData,df2=ProfData))

## End(Not run)
```

returnTextAreaInput	<i>Return message when the filter formula is not correct (mRNA > 500)</i>
---------------------	--

Description

Return message when the filter formula is not correct (mRNA > 500)

Usage

```
returnTextAreaInput(inputId,
  label= NULL,
  rows = 2,
  placeholder = NULL,
  resize= "vertical",
  value = "")
```

Arguments

inputId	The ID of the object
label	Text describes the box area
rows	Number of rows
placeholder	Error message if needed
resize	orientation of text
value	default text in the area box

Value

text message

Examples

```
ShinyApp <- 1
## Not run:
returnTextAreaInput(inputId = "data-filter",
                    label = "Error message",
                    rows = 2,
                    placeholder = "Provide a filter (e.g., Genes == 'ATM') and press return",
                    resize = "vertical",
                    value="")

## End(Not run)
```

Studies_obj *get object for grViz. Link Studies to genes*

Description

get object for grViz. Link Studies to genes

Usage

```
Studies_obj(df)
```

Arguments

df data frame with gene classes

Value

grViz object. a data frame with Study attributes

Examples

```
Studies_obj(data.frame("col1", "col2", "col3", "col4", "col5", "col6"))
## Not run:
Genes ranking      class postProb exprsMeanDiff exprsUpDw
1 FANCF            1 brca_tcga 1.00000        179.9226        UP
2 MLH1            1 gbm_tcga 0.99703        256.3173        UP

## End(Not run)
```

switchButton	<i>A function to change the Original checkbox of rshiny into a nice true/false or on/off switch button No javascript involved. Only CSS code.</i>
--------------	---

Description

To be used with CSS script 'button.css' stored in a 'www' folder in your Shiny app folder

Usage

```
switchButton(inputId, label = NULL, value = FALSE, col = "GB",
             type = "TF")
```

Arguments

inputId	The input slot that will be used to access the value.
label	Display label for the control, or NULL for no label.
value	Initial value (TRUE or FALSE).
col	Color set of the switch button. Choose between "GB" (Grey-Blue) and "RG" (Red-Green)
type	Text type of the button. Choose between "TF" (TRUE - FALSE), "OO" (ON - OFF) or leave empty for no text.

UnifyRowNames	<i>Unify row names in data frame with the same order of gene list.</i>
---------------	--

Description

Unify row names in data frame with the same order of gene list.

Usage

```
UnifyRowNames(x, geneList)
```

Arguments

x	data frame with gene symbol in the row name
geneList	a gene list

Value

a data frame having the gene in row name ordered as in gene list.

Examples

```
cgds <- CGDS("http://www.cbioportal.org/")
## Not run:
geneList <- whichGeneList("73")
ProfData <- getProfileData(cgds,
  geneList, "gbm_tcga_pub_mrna", "gbm_tcga_pub_all")
rownames(ProfData) <- NULL
geneListOrder <- UnifyRowNames(list(
  list1=list(df1=ProfData,df2=ProfData),
  list2=list(df3=ProfData,df4=ProfData)),
  geneList)

## End(Not run)
```

user_CNA

Example of Copy Number Alteration (CNA) dataset

Description

Example of Copy Number Alteration (CNA) dataset

Usage

user_CNA

Format

An object of class `data.frame` with 579 rows and 13 columns.

Author(s)

Karim Mezhoud <kmezhoud@gmail.com>

user_MetHM27

Example of Methylation HM27 dataset

Description

Example of Methylation HM27 dataset

Usage

user_MetHM27

Format

An object of class `data.frame` with 600 rows and 13 columns.

Author(s)

Karim Mezhoud <kmezhoud@gmail.com>

user_MetHM450	<i>Example of Methylation HM450 dataset</i>
---------------	---

Description

Example of Methylation HM450 dataset

Usage

user_MetHM450

Format

An object of class `data.frame` with 10 rows and 13 columns.

Author(s)

Karim Mezhoud <kmezhoud@gmail.com>

user_mRNA	<i>Example of mRNA expression dataset</i>
-----------	---

Description

Example of mRNA expression dataset

Usage

user_mRNA

Format

An object of class `data.frame` with 307 rows and 13 columns.

Author(s)

Karim Mezhoud <kmezhoud@gmail.com>

user_Mut	<i>Example of Mutation dataset</i>
----------	------------------------------------

Description

Example of Mutation dataset

Usage

```
user_Mut
```

Format

An object of class `data.frame` with 37 rows and 23 columns.

Author(s)

Karim Mezhoud <kmezhoud@gmail.com>

whichGeneList	<i>Verify which gene list is selected</i>
---------------	---

Description

Verify which gene list is selected

Usage

```
whichGeneList(geneListLabel)
```

Arguments

`geneListLabel` The label of GeneList. There are three cases: "Genes" user gene list, "Reactome_GeneList" GeneList plus genes from reactomeFI "file name" from Examples

Value

Gene List label

Examples

```
How <- "runManually"  
## Not run:  
whichGeneList("102")  
  
## End(Not run)
```

widgetThumbnail	<i>Capture html output widget as .png in R</i>
-----------------	--

Description

Capture html output widget as .png in R

Usage

```
widgetThumbnail(p, thumbName, width = 1024, height = 1024)
```

Arguments

p	is the html widget
thumbName	is the name of the new png file
width	1024
height	1024

Value

3 files .html, .js and .png

Examples

```
How <- "runManually"
## Not run:
# Load package
library(networkD3)
library(htmlwidgets)
# Create fake data
src <- c("A", "A", "A", "A", "B", "B", "C", "C", "D")
target <- c("B", "C", "D", "J", "E", "F", "G", "H", "I")
networkData <- data.frame(src, target)
# Plot
plot = simpleNetwork(networkData)
# Save html as png
widgetThumbnail(p = plot, thumbName = "plot", width = 1024, height = 1024)

## End(Not run)
```


Index

*Topic **datasets**

- epiGenomics, [10](#)
 - user_CNA, [29](#)
 - user_MethM27, [29](#)
 - user_MethM450, [30](#)
 - user_mRNA, [30](#)
 - user_Mut, [31](#)
- attriColorGene, [3](#)
attriColorValue, [3](#)
attriColorVector, [4](#)
attriShape2Gene, [5](#)
attriShape2Node, [5](#)
- bioCancer, [6](#)
bioCancer-package (bioCancer), [6](#)
- checkDimensions, [6](#)
coffeewheel, [7](#)
coffeewheelOutput, [8](#)
- displayTable, [8](#)
- Edges_Diseases_obj, [9](#)
epiGenomics, [10](#)
- findPhantom, [10](#)
- getFreqMutData, [11](#)
getGenesClassification, [11](#)
getList_Cases, [13](#)
getList_GenProfs, [14](#)
getListProfData, [12](#)
getMegaProfData, [14](#)
getSequenced_SampleSize, [15](#)
grepRef, [16](#)
- metabologram, [17](#)
metabologramOutput, [18](#)
Mutation_obj, [18](#)
- Node_df_FreqIn, [19](#)
Node_Diseases_obj, [20](#)
Node_obj_CNA_ProfData, [20](#)
Node_obj_FreqIn, [21](#)
- Node_obj_Met_ProfData, [22](#)
Node_obj_mRNA_Classifier, [22](#)
- renderCoffeewheel, [23](#)
renderMetabologram, [24](#)
reStrColorGene, [24](#)
reStrDimension, [25](#)
reStrDisease, [26](#)
returnTextAreaInput, [26](#)
- Studies_obj, [27](#)
switchButton, [28](#)
- UnifyRowNames, [28](#)
user_CNA, [29](#)
user_MethM27, [29](#)
user_MethM450, [30](#)
user_mRNA, [30](#)
user_Mut, [31](#)
- whichGeneList, [31](#)
widgetThumbnail, [32](#)